Package ‘gamlss.ggplots’

November 12, 2022

Description Plotting functions for Generalised Additive Models for Location Scale and Shape.

Title Plotting Generalised Additive Model for Location, Scale and Shape

LazyLoad yes

Version 2.0-1

Date 2022-11-12

Depends R (>= 3.5.0), gamlss.dist, gamlss (>= 4.3.3), ggplot2, gamlss.foreach

Imports methods, ggridges, ellipse, gamlss.inf

Suggests glmnet

Author Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.ukg`>, Bob Rigby, Fernanda De Bastiani, Julian Merder

Maintainer Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>

URL `https://www.gamlss.com/`

NeedsCompilation no

Repository CRAN

Date/Publication 2022-11-12 16:50:02 UTC

R topics documented:

`gamlss.ggplots-package` *Plotting Generalised Additive Model for Location, Scale and Shape*

Description

Plotting functions for Generalised Additive Models for Location Scale and Shape.

Details

The DESCRIPTION file:

Package:	gamlss.ggplots
Description:	Plotting functions for Generalised Additive Models for Location Scale and Shape.
Title:	Plotting Generalised Additive Model for Location, Scale and Shape
LazyLoad:	yes
Version:	2.0-1
Date:	2022-11-12
Depends:	R (>= 3.5.0), gamlss.dist, gamlss (>= 4.3.3), ggplot2, gamlss.foreach
Imports:	methods, ggridges, ellipse, gamlss.inf
Suggests:	glmnet
Author:	Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.ukg>, Bob Rigby, Fernanda De Bastiani, Julian M
Maintainer:	Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>
License:	GPL-2 | GPL-3
URL:	https://www.gamlss.com/
NeedsCompilation:	no
Packaged:	2022-02-06 19:52:24 UTC; dimitriosstasinopoulos

Index of help topics:

`boot_coef`	`Plotting Bootstrap Coefficients`
`centile_bucket`	`Centile bucket plot`
`family_pdf`	`Plotting Probabilities Density Functions (pdf's) for GAMLSS`
`fitted_cdf`	`Plotting Cumulative Distribution Functions (cdf's) for GAMLSS,`
`fitted_centiles`	`Plotting centile (growth) curves`
`fitted_devianceIncr`	`Plotting the deviance increment of GAMLSS`
`fitted_leverage`	`Plot of the linear leverage of a GAMLSS model`
`fitted_terms`	`Plotting fitted additive terms`
`gamlss.ggplots-package`	`Plotting Generalised Additive Model for Location, Scale and Shape`
`histSmo_plot`	`Supporting histSmo()`
`model_GAIC`	`Plotting GAIC for GAMLSS models`
`model_pca`	`Plotting residuals using PCA`
`moment_bucket`	`Moment bucket plot`
`moment_gray_half`	`Functions to create the background for the bucket plots`
`pcr_coef_path`	`Plotting the fitted path of a PCR model.`
`pe_param`	`Partial Effect of a term on the parameters and predictors`
`resid_density`	`Density of the residuals in a GAMLLSS model`
`resid_dtop`	`Detrended Transformed Owen's Plot and ECDF for the residuals`
`resid_index`	`A residual plots`
`resid_qqplot`	`QQ-plot of the residuals of a GAMLSS model`
`resid_symmetry`	`Symmetry plots`
`resid_wp`	`Worm plot using ggplot2`
`resp_mu`	`Plotting the response against quantities of the fitted model`
`y_hist`	`Histogram and density plot.`

The following convention has been used to name the functions:

`fitted_NAME`: plots concerning fitted values from a single fitted model

`resid_NAME`: plots concerning residuals from a single fitted model

`predict_NAME`: plots concerning prediction values from a single fitted model usually having `newdata` option.

`model_NAME`: plots concerning different fitted models

where `NAME` refer to different characteristics.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.ukg>, Bob Rigby, Fernanda De Bastiani, Julian Merder

Maintainer: Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC, doi:10.1201/9780429298547. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, doi:10.18637/jss.v023.i07.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M. D., Rigby, R. A., and De Bastiani F., (2018) GAMLSS: a distributional regression approach, *Statistical Modelling*, Vol. **18**, pp, 248-273, SAGE Publications Sage India: New Delhi, India.

(see also `https://www.gamlss.com/`).

See Also

`gamlss`, `gamlss.family`

Examples

```
library(gamlss)
m1 <- gamlss(y~pb(x), data=abdom)
resid_index(m1)
```

`boot_coef` *Plotting Bootstrap Coefficients*

Description

The function plots in one or multiple pages the results from a boostrap generated by the function `BayesianBoot()`, `NonParamatricBoot` or `nonpar_boot()`.

Usage

```
boot_coef(x, terms = NULL, hist.col = "black",
                hist.fill = "white", dens.fill = "#FF6666",
                alpha = 0.2, nrow = NULL, ncol = NULL,
                plots.per.page = 9, one.by.one = FALSE, title, ...)
```

Arguments

`x`	a `Bayesian.boot` or `NonParametric.boot` object
`terms`	which terms to plot (default NULL means all terms
`hist.col`	colour of the border histogram
`hist.fill`	the colour of the histogram
`dens.fill`	the colour of the density estimate
`alpha`	transparity constant
`nrow`	how namy rows
`ncol`	how many columns
`plots.per.page`	
`one.by.one`	the maxiimum plots per page
`title`	the title
`...`	for more argument

Details

The function plots in one ore multiple pages the results from a boostrap simulation

Value

Greates a ggplot object

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`term.plot`

Examples

```
data(aids)
a<-gamlss(y~pb(x)+qrt,data=aids,family=NBI)
registerDoParallel(cores = 2)
B1 <- BayesianBoot(a, B=100)
stopImplicitCluster()
boot_coef(B1)
```

`centile_bucket` *Centile bucket plot*

Description

A bucket plot is a graphical way to check the skewness and kurtosis of a continuous variable or the residuals of a fitted GAMLSS model. It plots the centile skewness (tail or central) and transformed centile kurtosis of the variable (or residuals) together with a cloud of points obtained using a non-parametric bootstrap from the original variable (or residuals). It also provides a graphical way of performing a Monte Carlo simulation test on whether the centile skewness and transformed centile kurtosis of the variable of interest are simultaneously equal to zero.

There are two function here:

i) `cenlile_bucket()` for a single bucket plot. Note that `model_cen_bucket()` and `centile_bucket()` are synonymous.

ii) `centile_bucket_wrap()` for multiple bucket plots cut according to terms in the model.

Usage

```
centile_bucket(x, ..., type = c("tail", "central"), weights = NULL,
       no_bootstrap = 99, col_bootstrap = hcl.colors(length.obj,
       palette = "Set 2"), alpha_bootstrap = 1, text_to_show = NULL,
       cex_text = 5, col_text = "black", colour_bucket = FALSE,
       line_width = 0.5, sim_test = FALSE, no_sim_test = 1000,
       col_sim_test = gray(0.7), alpha_sim_test = 0.1, seed_test = 1234)

model_cent_bucket(x, ..., type = c("tail", "central"), weights = NULL,
       no_bootstrap = 99, col_bootstrap = hcl.colors(length.obj,
       palette = "Set 2"), alpha_bootstrap = 1, text_to_show = NULL,
       cex_text = 5, col_text = "black", colour_bucket = FALSE,
       line_width = 0.5, sim_test = FALSE, no_sim_test = 1000,
       col_sim_test = gray(0.7), alpha_sim_test = 0.1, seed_test = 1234)


centile_bucket_wrap(x, ..., type = c("tail", "central"), weights = NULL,
      xvar = NULL, n_inter = 4, no_bootstrap = 99,
      col_bootstrap = hcl.colors(length.obj, palette = "Set 2"),
```

```
    alpha_bootstrap = 1, text_to_show = NULL, check_overlap_text = FALSE,
    cex_text = 5, col_text = "black", colour_bucket = FALSE,
    line_width = 0.5, sim_test = FALSE, no_sim_test = 1000,
    col_sim_test = gray(0.7), alpha_sim_test = 0.1, seed_test = 1234)
```

Arguments

`x`	x should be a continuous vector of a GAMLSS fitted model.
`...`	for more that one continuous vectors or fitted models
`type`	whether "tail" of "central" skewness and kurtosis
`weights`	if priors weights are needed
`no_bootstrap`	the number of bootstrap samples for the cloud around the point of skewness and kurtosis.
`col_bootstrap`	The colour of the bootstrap samples
`alpha_bootstrap`	The transparency parameter of the bootstrap samples.
`text_to_show`	what text to show in the plots, default the names of vectors or models
`cex_text`	the character size of the text
`col_text`	the colour of the text
`colour_bucket`	whether colour or gray lines in the bucket
`line_width`	the line width
`sim_test`	whether to Monde Carlo simulation is needed to check the null hypothesis that there is no centile skewness and transformed centile kurtosis in the sample.
`no_sim_test`	The number of simulation for the test
`col_sim_test`	the colour used for displaying the Monde Carlo test values
`alpha_sim_test`	The transparency parameter of the Monde Carlo samples.
`seed_test`	A seed value for the Monde Carlo simulation.
`xvar`	the x term
`n_inter`	how many intervals needed
`check_overlap_text`	whether to check overlapping text

Details

More details about centile bucket plots is given in De Bastiani *et al.* (2022)

Value

A plot displaying the centile skewness and transformed centile kurtosis of the sample or residual of a model.

Note

The bucket plot provides an additional residual diagnostic tool that can be used for fitted model checking, alongside other diagnostic tools, for example worm plots, and Q (and Z) statistics.

Author(s)

Mikis Stasinopoulos, <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby <`r.rigby@londonmet.ac.uk`> and Fernanda De Bastiani

References

De Bastiani, F., Stasinopoulos, D. M., Rigby, R. A., Heller, G. Z., and Lucas A. (2022) Bucket Plot: A Visual Tool for Skewness and Kurtosis Comparisons. To be published.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/9780429298547 An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, doi:10.18637/jss.v023.i07.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M. D., Rigby, R. A., and De Bastiani F., (2018) GAMLSS: a distributional regression approach, *Statistical Modelling*, Vol. **18**, pp, 248-273, SAGE Publications Sage India: New Delhi, India.

(see also `https://www.gamlss.com/`).

See Also

`wp`, `Q.stats`

Examples

```
m1 <- gamlss(R~pb(Fl)+pb(A), data=rent, family=GA)
centile_bucket(m1)
## Not run:
centile_bucket_wrap(m1, xvar=rent$A)

## End(Not run)
```

`family_pdf` *Plotting Probabilities Density Functions (pdf's) for GAMLSS*

Description

The function `family_pdf()` takes a GAMLSS family distribution and plots different pdf's according to the specified parameters.

The function `fitted_pdf()` takes a gamlss fitted object and plots the fitted distributions for specified observations.

The function `fitted_pdf_data()` it does the same as `fitted_pdf()` but it adds also the observation values as grey vertical lines.

The function `predict_pdf()` takes a fitted object and test data and plots the predictive pdf's.

Usage

```
family_pdf(family = NO(), mu = NULL, sigma = NULL, nu = NULL, tau = NULL,
         title, from = 0, to = 10, no.points = 201,
         alpha = 0.4, col.fill = hcl.colors(lobs, palette = "viridis"),
         size.seqment = 1.5, plot.point = TRUE, size.point = 1,
         plot.line = TRUE, size.line = 0.2, ...)

fitted_pdf(model, obs, title, from = 0, to = 10, no.points = 201, alpha = 0.4,
         col.fill = hcl.colors(lobs, palette = "viridis"),
        size.seqment = 1.5, plot.point = TRUE, size.point = 1, plot.line = TRUE,
         size.line = 0.2, ...)

fitted_pdf_data(model, obs, from, to, ...)

predict_pdf(model, newdata, title, from = 0, to = 10, no.points = 201,
         alpha = 0.4, col.fill = hcl.colors(lobs, palette = "viridis"),
         size.seqment = 1.5, plot.point = TRUE, size.point = 1,
         plot.line = TRUE, size.line = 0.2, ...)
```

Arguments

`family`	A GAMLSS family
`model`	A GAMLSS fitted model
`obs`	observations to plot fitted distributions
`newdata`	for test data
`mu`	the `mu` parameter value(s)
`sigma`	the `sigma` parameter value(s)
`nu`	the `nu` parameter value(s)
`tau`	the `tau` parameter value(s)

`title`	a diferent title for the default
`from`	minimum value for the response
`to`	maximum value for the response
`no.points`	number of points (relevant for continuous responses)
`alpha`	trasparency factor
`col.fill`	the colour pallet default is `hcl.colors(lobs, palette="viridis")`
`size.seqment`	for discrete responses the size of the bars
`plot.point`	for discrete responses whether to put poits on the top of the bars
`size.point`	for discrete responses
`plot.line`	for discrete responses whether to joint the bars with lines
`size.line`	for discrete responses the size of the joining lines
`...`	for extra argumnets

Details

The functions `family_pdf()` and `fitted_pdf()` are ggplot version of the function `pdf.plot()` used to plot fitted distributions of GAMLSS family at specified observation values. Note that the range of the response has to be specified using the argument `from` `to`.

For discrete fitted distributions maybe increase the value of `alpha` for clearer plot.

For binomial type of data (discrete response with upper limit) the function `family_pdf()` takes the argument `to` as the binomial denominator, For fitted model with binomial type responses the function `fitted_pdf()` takes the binomial denominator form the fitted model and set the argument `to` to the maximum of those binomial denominators.

Value

Creates a plot

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

gamlss, resid_density

Examples

```
####################################
# function fitted_pdf
# continuous variabe
a1 <- gamlss(y~pb(x),sigma.fo=~pb(x), data=abdom, family=LO)
fitted_pdf(a1, obs=c(10,15,20), from=30, to=100)
# count data
p1 <- gamlss(y~pb(x)+qrt, data=aids, family=NBI)
fitted_pdf(p1, obs=c(10:15), from=25, to=130, alpha=.9)
# binomial type
h<-gamlss(y~ward+loglos+year, sigma.formula=~year+ward, family=BB, data=aep)
fitted_pdf(h, obs=c(10:15),  alpha=.9)
####################################
# function predict_pdf
predict_pdf(a1, newdata=abdom[10:20, ], from=30, to=100)
# count data
predict_pdf(p1, newdata=aids[10:15, ], from=30, to=150)
# binomial
predict_pdf(h, newdata=aep[10:15, ], from=0, to=20)
####################################
# function family_pdf
# continuous
family_pdf(from=-5,to=5, mu=0, sigma=c(.5,1,2))
# count data
family_pdf(NBI, to=15, mu=1, sigma=c(.5,1,2), alpha=.9, size.seqment = 3)
# binomial type
family_pdf(BB, to=15, mu=.5, sigma=c(.5,1,2),  alpha=.9, , size.seqment = 3)
```

`fitted_cdf` *Plotting Cumulative Distribution Functions (cdf's) for GAMLSS,*

Description

The function `family_cdf()` takes a GAMLSS family distribution and plots different pdf's according to the specified parameters.

The function `fitted_cdf()` takes a gamlss fitted object and plots the fitted distributions for specified observations.

The function `fitted_cdf_data()` is similat to `fitted_cdf()` but also adds the data points as gray vertical lines.

The function `predict_pdf()` (NOT IMPLEMENTED YET) takes a fitted object and test data and plots the predictive cdf's.

Usage

```
fitted_cdf(model, obs, title, from = 0, to = 10, no.points = 201,
          alpha = 1, size.line = 1.2,
          col.fill = hcl.colors(lobs, palette = "viridis"),
          size.seqment = 1.5, size.point = 1,
          plot.line = TRUE, size.line.disc = 0.2, lower.tail = TRUE, ...)

fitted_cdf_data(model, obs, from, to, ...)

predict_cdf(model, newdata, title, from = 0, to = 10, no.points = 201,
          alpha = 0.4, size.line = 1.2,
          col.fill = hcl.colors(lobs, palette = "viridis"),
          size.seqment = 1.5, plot.point = TRUE, size.point = 1,
          plot.line = TRUE, size.line.disc = 0.2, lower.tail = TRUE, ...)

family_cdf(family = NO(), mu = NULL, sigma = NULL, nu = NULL,
         tau = NULL, title, from = 0, to = 10, no.points = 201,
         alpha = 0.4, size.line = 1.2, col.fill = hcl.colors(lobs,
         palette = "viridis"), size.seqment = 1.5, plot.point = TRUE,
         size.point = 1, plot.line = TRUE, lower.tail = TRUE, ...)
```

Arguments

`family`	A GAMLSS family
`model`	A GAMLSS fitted model
`obs`	observations to plot fitted distributions
`newdata`	for test data
`mu`	the `mu` parameter value(s)
`sigma`	the `sigma` parameter value(s)
`nu`	the `nu` parameter value(s)
`tau`	the `tau` parameter value(s)
`title`	a diferent title for the default
`from`	minimum value for the response
`to`	maximum value for the response
`no.points`	number of points (relevant for continuous responses)
`alpha`	trasparency factor
`col.fill`	the colour pallet default is `hcl.colors(lobs, palette="viridis")`
`size.seqment`	for discrete responses the size of the bars
`plot.point`	for discrete responses whether to put poits on the top of the bars
`size.point`	for discrete responses
`plot.line`	for discrete responses whether to joint the bars with lines
`size.line.disc`	for discrete responses the size of the joining lines

`size.line`	The size of the lines
`lower.tail`	if `TRUE` cdf is plotted if `FALSE` the survival function
`...`	for extra argumnets

Details

The functions `family_cdf()`, `fitted_cdf()`, and `predict_cdf()` are function to plot cdf's for a `gamlss.family`, fitted `gamlss` model or predictive `gamlss` model, respectively.

For discrete fitted distributions maybe increase the value of `alpha` for clearer plot.

For binomial type of data (discrete response with upper limit) the function `family_cdf()` takes the argument `to` as the binomial denominator, For fitted model with binomial type responses the function `fitted_cdf()` takes the binomial denominator form the fitted model and set the argument `to` to the maximum of those binomial denominators.

Value

Creates a plot

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`gamlss`

Examples

```
# function fitted_cdf
# continuous variabe
a1 <- gamlss(y~pb(x),sigma.fo=~pb(x), data=abdom, family=LO)
fitted_cdf(a1, obs=c(10,15,20), from=30, to=100)
fitted_cdf(a1, obs=c(10,15,20), from=30, to=100, lower.tail=FALSE)
# count data
p1 <- gamlss(y~pb(x)+qrt, data=aids, family=NBI)
```

```
fitted_cdf(p1, obs=c(10:15), from=10, to=130, alpha=.9)
fitted_cdf(p1, obs=c(10:15), from=10, to=130, alpha=.9, lower.tail=FALSE)
# binomial type
h<-gamlss(y~ward+loglos+year, sigma.formula=~year+ward, family=BB, data=aep)
fitted_cdf(h, obs=c(10:15),  alpha=.9)
fitted_cdf(h, obs=c(10:15),  alpha=.9, lower.tail=FALSE)
###################################
# function predict_pdf
predict_cdf(a1, newdata=abdom[c(10,15,20), ], from=30, to=100)
predict_cdf(a1, newdata=abdom[10:20, ], from=30, to=100, lower.tail=FALSE)
# count data
predict_cdf(p1, newdata=aids[10:15, ], from=10, to=150)
predict_cdf(p1, newdata=aids[10:15, ], from=10, to=150, lower.tail=FALSE)
# binomial
predict_cdf(h, newdata=aep[10:15, ], from=0, to=20)
predict_cdf(h, newdata=aep[10:15, ], from=0, to=20, lower.tail=FALSE)
###################################
# function family_cdf
# continuous
family_cdf(from=-5,to=5, mu=0, sigma=c(.5,1,2))
family_cdf(from=-5,to=5, mu=0, sigma=c(.5,1,2), lower.tail=FALSE)
# count data
family_cdf(NBI, to=15, mu=1, sigma=c(.5,1,2), alpha=.9, size.seqment = 3)
family_cdf(NBI, to=15, mu=1, sigma=c(.5,1,2), alpha=.9, size.seqment = 3, lower.tail=FALSE)
# binomial type
family_cdf(BB, to=15, mu=.5, sigma=c(.5,1,2),  alpha=.9, , size.seqment = 3)
family_cdf(BB, to=15, mu=.5, sigma=c(.5,1,2),  alpha=.9, , size.seqment = 3, lower.tail=FALSE)
```

`fitted_centiles` *Plotting centile (growth) curves*

Description

The function `fitted_centiles()` plots centiles curves for distributions belonging to the GAMLSS family of distributions. The plot is equivalent to the standard plot of `gamlss:::centiles()` without a legend.

The function `fitted_centiles_legend()` plots centiles curves for distributions belonging to the GAMLSS family of distributions and it is equivalent to the standard plot of `gamlss:::centiles()` with a legend. The function is slower than `fitted_centiles()` since in order to plot the legend the data have to expanded.

The function `model_centiles()` plots centile curves for more than one model. There is no equivanlt standard plot but it perform the same function as `gamlss:::centiles.com()` which compares centiles from different models.

Usage

```
fitted_centiles(obj, xvar,
                cent = c(99.4, 98, 90, 75, 50, 25, 10, 2, 0.4),
                points = TRUE, point.col = "gray",
```

```
                point.size = 1, line.size = 0.8,
                line.col = hcl.colors(lc, palette = "Dark 2"),
                xlab = NULL, ylab = NULL, title, ...)

fitted_centiles_legend(obj, xvar,
                cent = c(99.4, 98, 90, 75, 50, 25, 10, 2, 0.4),
                points = TRUE, point.col = "gray", point.size = 1,
                line.size = 0.8, line.col = hcl.colors(ncent,
                palette = "Dark 2"),
                show.legend = TRUE, save.data = FALSE, title,
                xlab = NULL, ylab = NULL, ...)

model_centiles(obj, ..., cent = c(97, 90, 75, 50, 25, 10, 3),
                xvar, xlab = "age", points = TRUE,
                point.col = gray(0.8),
                point.size = 0.05, line.size = 0.7,
                line.col = hcl.colors(ncent,palette = "Dark 2"),
                ncol = 2, nrow = ceiling(nnames/ncol),  in.one = FALSE,
                title)
```

Arguments

`obj`	a fitted gamlss object
`xvar`	the (unique) explanatory variable
`cent`	a vector with elements the % centile values for which the centile curves have to be evaluated
`points`	whether to plot the points (`TRUE`) of the data or not (`FALSE`)
`point.col`	the colour of the points
`point.size`	the zize of the points
`line.size`	the sized of the centile lines
`line.col`	the colour of the centile lines
`xlab`	the label of the x-axis variable
`ylab`	the label of the resposnse variable
`in.one`	whether the `model_centile` plot should ne one or multiple
`title`	the title if need it otherwise a dfault title is pronted
`show.legend`	whether to show the legend
`save.data`	whether to save the `data.frame` of the plot
`nrow`	the number of rows in the `model_centiles()` plot
`ncol`	the number of columns in the `model_centiles()` plot
`...`	for extra arguments for `fitted_centiles()`, and `fitted_centiles.legend()` and extra models for `model_centiles()`

Details

Centiles are calculated using the fitted values in `obj` and `xvar` must correspond exactly to the predictor in `obj` to plot correctly.

Value

A plot is created

Warning

This function is appropriate only when one continuous explanatory variable is fitted in the model

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda de Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. (see also `https://www.gamlss.com/`).

See Also

`centiles`

Examples

```
data(abdom)
h<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCTo, data=abdom)
h1 <- gamlss(y~pb(x), sigma.formula=~pb(x), family=LO, data=abdom)
fitted_centiles(h)
fitted_centiles_legend(h)
model_centiles(h, h1)
```

`fitted_devianceIncr` *Plotting the deviance increment of GAMLSS*

Description

There are two plotting function here:

i) `fitted_devianceIncr()` plots the fitted model deviance components. This is useful for identifying observations with unusual y-values (given the current fitted distribution).

iii) `model_devianceIncr_diff` plots the difference of deviance increments from two fitted GAMLSS model. This function is useful if the GAIC and the residuals contradict each other. For example the GAIC is better for model 1 but the residuals look lot better for model 2. This can happens if the two distributions are better suited to fit different parts of the response distribution i.e. one model fits the center better but the other fits the tail better.

Usage

```
fitted_devianceIncr(obj, plot = TRUE, title, quan.val = 0.99,
         annotate = TRUE, line.col = "steelblue4",
         point.col = "darkblue", annot.col = "white",
         newdata = NULL)


model_devianceIncr_diff(model1, model2, against = "index",
         tol = 20, data, newdata)
```

Arguments

`obj`	a GAMLSS fitted object
`plot`	whether to create just the plot or save also the values with high deviance increment
`title`	a tittle if needed it.
`quan.val`	The quantile values of the deviance increment from which the obsrevrvarion should be identify
`annotate`	whether to plot the quantile values above in the plot.
`line.col`	the colour of the line
`point.col`	the colour of the points
`annot.col`	the colour of the annotation for the deviance increment plot
`model1`	The first fitted GAMLSS model
`model2`	The second fitted GAMLSS model
`against`	you can plot the deviance increment an index, the response or an x-variable

`tol`	if the absolute value of deviance increment exceeds the `tol` the number of the observation is plotted
`data`	The data if can not be found from `model1`
`newdata`	evaluates the function in new data

Details

The functions are diagnostic tools to check unusual observations in the response.

Value

return a plot

Author(s)

Mikis Stasinopulos, Rober Rigby and Fernanda de Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`gamlss`

Examples

```
m1 <- gamlss(R~pb(Fl)+pb(A)+H+loc, data=rent, family=GA )
m2 <- gamlss(R~pb(Fl)+pb(A)+H+loc, data=rent, family=NO )
fitted_devianceIncr(m1)
model_devianceIncr_diff(m1,m2, against="Fl")
```

`fitted_leverage` *Plot of the linear leverage of a GAMLSS model*

Description

This is plot of the "linear" leverage of a GAMLSS fitted model. By linear we mean the leverage (hat-values) we would have obtain in all the explanatory variables for all distribution parameters where put together and used to fit a linear model to the response. The "linear" leverage is them the hat-values obtained by fitting this simple linear model. Hopefully the "linear" leverage can indicate observations with extreme values in the x's. Note that observations with hight linear leverage may not be influential in the GAMLSS fitting especially if the x-variables are fitted using smoothers.

Usage

```
fitted_leverage(obj, plot = TRUE, title, quan.val = 0.99,
          annotate = TRUE, line.col = "steelblue4",
          point.col = "steelblue4", annot.col = "darkred")
```

Arguments

`obj`	A GAMLSS fitted model
`plot`	whether to plot ot not
`title`	for different title than the default
`quan.val`	which quantile value of the leverage should be taked to indicate the observation values
`annotate`	whether to annotate the extreme levarages
`line.col`	the colour of the lines
`point.col`	the colout of the points
`annot.col`	the colour used for annotation

Value

Returns a plot of the linear leverage against index.

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`gamlss`

Examples

```
m1 <- gamlss(R~pb(Fl)+pb(A)+loc+H, data=rent, family=GA)
fitted_leverage(m1)
```

`fitted_terms` *Plotting fitted additive terms*

Description

The function `fitted_terms()` is doing what the function `term.plot()` is doing for GAMLSS models but it uses `ggplot2` package. The function `pe_terms()` is synonymous to `fitted_terms()`

Usage

```
fitted_terms(object,
          what = c("mu", "sigma", "nu", "tau"),
          parameter = NULL, data = NULL, terms = NULL,
          envir = environment(formula(object)),
          partial = FALSE, rug = FALSE, rug.sides = "b",
          rug.col = "gray",  alpha = 0.2,
          ylim = c("common", "free"), xlabs = NULL,
          ylabs = NULL, main = NULL, term.col = "darkred",
          resid.col = "lightblue", resid.alpha = 0.8,
          resid.size = 1, nrow = NULL, ncol = NULL,
          plots.per.page = 9, one.by.one = FALSE,
          surface.gam = FALSE, polys = NULL,
          polys.scheme = "topo", col.ribbon = "darksalmon",
          col.shaded = "gray", ...)

pe_terms(object,
          what = c("mu", "sigma", "nu", "tau"),
          parameter = NULL, data = NULL, terms = NULL,
          envir = environment(formula(object)),
          partial = FALSE, rug = FALSE, rug.sides = "b",
          rug.col = "gray",  alpha = 0.2,
```

```
             ylim = c("common", "free"), xlabs = NULL,
             ylabs = NULL, main = NULL, term.col = "darkred",
             resid.col = "lightblue", resid.alpha = 0.8,
             resid.size = 1, nrow = NULL, ncol = NULL,
             plots.per.page = 9, one.by.one = FALSE,
             surface.gam = FALSE, polys = NULL,
             polys.scheme = "topo", col.ribbon = "darksalmon",
             col.shaded = "gray", ...)
```

Arguments

`object`	a gamlss fitted model
`what`	which distribution parameters, i.e. "mu"
`parameter`	which distribution parameters (equivalent to `what`)
`data`	data frame in which variables in `object` can be found
`terms`	which terms to plot (default NULL means all terms); a vector passed to lpred(..., type = "terms", terms =*)
`envir`	environment in which variables in `object` can be found
`partial`	logical; should partial residuals be plotted?
`rug`	add rug plots to the axes
`rug.sides`	which side the rug "b"=bottom
`rug.col`	the colour for the rug
`alpha`	the alpha for the shade
`ylim`	there are two options here a) "common" and b) "free". The "common"" option plots all figures with the same ylim range and therefore allows the viewer to check the relative contribution of each terms compate to the rest. In the ‘free’ option the limits are computed for each plot seperatly.
`xlabs`	the x label
`ylabs`	the y label
`main`	title NOT WORKING
`term.col`	the colour of the line for term
`resid.col`	the colour of the partial residuals
`resid.alpha`	The alpha for the partial residuals
`resid.size`	the size of the partial residuals
`nrow`	the number or rows in a mupliple plot
`ncol`	the number of rows in a mupliple plot
`plots.per.page`	the number of plots per page in multiple plots
`one.by.one`	whether to plot the terms one by one
`surface.gam`	whether to use surface plot if a ga() term is fitted
`polys`	for GRMF models

`polys.scheme`	The polygone information file for MRF models
`col.ribbon`	he colour of the ribbon
`col.shaded`	he colour of the shaded area
`...`	for extra argument

Value

A multiple plot

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`term.plot`

Examples

```
data(aids)
a<-gamlss(y~pb(x)+qrt,data=aids,family=NBI)
fitted_terms(a, pages=1)
rm(a)
```

`histSmo_plot` *Supporting histSmo()*

Description

This function helps to plot density estimates created by the `histSmo()` function.

Usage

```
histSmo_plot(x, col_fill_bar = gray(0.5), col_bar = "pink",
        col_line = "darkblue", width_line = 1, title, xlabel)
```

Arguments

`x`	a `histSmo` object
`col_fill_bar`	The fill colour of the bars
`col_bar`	the colour of the border of thebars
`col_line`	the colour of the lines
`width_line`	the width of the lines
`title`	title if needed
`xlabel`	x axis lable if needed.

Details

This function supports `histSmo()`.

Value

A plot

Author(s)

Mikis Stasinopulos, Rober Rigby and Fernanda de Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also https://www.gamlss.com/).

See Also

`histSmo`

Examples

```
a1 <-histSmo(abdom$y)
gg1 <-histSmo_plot(a1)
gg1
```

`model_GAIC` *Plotting GAIC for GAMLSS models*

Description

The function `model_GAIC()` is similar to the function `GAIC.scaled()` of the package **gamlss**. It produces, [for a given set of different fitted models or for a table produced by `chooseDist()`], the scaled Akaike values (see Burnham and Anderson (2002) section 2.9 for a similar concept of the GAIC weights. The plot of the GAIC's should not be interpreted as posterior probabilities of models given the data but can be used for model selection purpose since they produce a scaled ranking of the model using their relative importance i.e. from the worst to the best model.

The function `model_GAIC_lollipop()` is almost identical to `model_GAIC()` but the result is a lollipop plot.

Usage

```
model_GAIC(object, ..., k = 2, c = FALSE, plot = TRUE,
       which = 1, diff.dev = 1000, text.to.show = NULL,
       col = "rosybrown", width = 0.9, horiz = TRUE,
       scale = c("[0,1]","[max,min]"), title)

model_GAIC_lollipop(object, ..., k = 2, c = FALSE, plot = TRUE,
       which = 1, diff.dev = 1000, text.to.show = NULL,
       col = "rosybrown", width = 0.9, horiz = TRUE,
       scale = c("[0,1]", "[max,min]"), order.val = TRUE, title)
```

Arguments

`object`	a set of `gamlss` fitted model(s) or a matrix table produced by `chooseDist()`.
`...`	it allows several GAMLSS object to be compared using a GAIC
`k`	the penalty with default k=2
`c`	whether the corrected AIC, i.e. AICc, should be used, note that it applies only when k=2
`plot`	whether to plot with default equal `TRUE`
`which`	which column of GAIC table to plot

`diff.dev`	this argument applies only a matrix table produced by `chooseDist()` and prevents models with a difference in deviance greater than `diff.dev` from the 'best' model to be considered (or plotted).
`text.to.show`	if NULL, `model_GAIC()` shows the model names otherwise the character in this list (the length of which should be equal to the length of models)
`col`	The colour of the bars
`width`	the width of the bars
`horiz`	whether to plot the bars horizontally (default) or vertically
`scale`	the scale of the plot, `"[0,1]"` plots the AIC's from the worst to the best models in a scale from [0,1]. `"[max,min]"` plots the AIC's from the worst model to the best model but in the original scale of the AIC's
`title`	if different title is needed
`order.val`	whether to order the models from the best to the worst

Details

The option allow the AIC to be plotted from worst to best on a $[0,1]$ scale using the formula i.e. $(AIC_w - AIC_m)/(AIC_w - AIC_b))$ where the AIC_w and AIC_b are the worst and best AIC, respectively, and AIC_m is the AIC of the current model. If the option `scale` is set to`[max,min]` the difference $(AIC_w - AIC_m)$ is plotted.

Value

It returns a bar plot using package **ggplot2**.

Author(s)

Mikis Stasinopoulos `<d.stasinopoulos@londonmet.ac.uk>`, Bob Rigby and Fernanda De Bastiani

References

Burnham K. P. and Anderson D. R (2002). *Model Selection and Multimodel Inference A Practical Information-Theoretic Approach*, Second Edition, Springer-Verlag New York, Inc.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

GAIC.scaled

Examples

```
data(abdom)
m1 <-  gamlss(y~x, family=NO, data=abdom)
m2 <-  gamlss(y~x, sigma.fo=~x, family=NO, data=abdom)
m3 <-  gamlss(y~pb(x), sigma.fo=~x, family=NO, data=abdom)
m4 <-  gamlss(y~pb(x), sigma.fo=~pb(x), family=NO, data=abdom)

model_GAIC(m1,m2, m3, m4)

MT <- chooseDist(m3)
model_GAIC(MT)
model_GAIC(MT, which=2)
model_GAIC_lollipop(m1,m2, m3, m4)
```

`model_pca` *Plotting residuals using PCA*

Description

The function `model_pca()` plots several GAMLSS residuals using Principal Component Analysis.

Usage

```
model_pca(obj, ..., scale = TRUE, arrow_size = 1.5)
```

Arguments

`obj`	A gamlss object
`...`	for extra GAMLSS models
`scale`	whether to scale the residuals
`arrow_size`	the arrow sizw in the biplot

Details

The function `model_pca()` plot a `biplot()` plot of the residuals from different models. It uses Principal Component Analysis in the residuals of different models and plots the resuls.

Value

A biplot of the first two components is plotted.

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also https://www.gamlss.com/).

See Also

`gamlss`, `resid_index`

Examples

```
m1 <- gamlss(y~x, data=abdom)
m2 <- gamlss(y~pb(x), data=abdom)
m3 <- gamlss(y~pb(x), sigma.fo=~pb(x), data=abdom)
model_pca(m1,m2,m3)
```

`moment_bucket` *Moment bucket plot*

Description

A bucket plot is a graphical way to check the skewness and kurtosis of a continuous variable or the residuals of a fitted GAMLSS model. It plots the transformed moment skewness and transformed moment kurtosis of the variable (or residuals) together with a cloud of points obtained using a non-parametric bootstrap from the original variable (or residuals). It also provides a graphical way of performing the Jarque-Bera test (JarqueandBera,1980).

There are two function here:

i) `moment_bucket()` for a single bucket plot. Note that `model_mom_bucket()` and `moment_bucket()` are synonymous.

ii) `moment_bucket_wrap()` for multiple bucket plots cut according to terms in the model.

Usage

```
moment_bucket(x, ..., weights = NULL, no_bootstrap = 99,
           col_bootstrap = hcl.colors(length.obj, palette = "Set 2"),
           alpha_bootstrap = 1, text_to_show = NULL,
           cex_text = 5, col_text = "black", colour_bucket = FALSE,
           line_width = 0.5, col_JB_test = gray(.7), alpha_JB_test = .1)
```

```
model_mom_bucket(x, ..., weights = NULL, no_bootstrap = 99,
          col_bootstrap = hcl.colors(length.obj, palette = "Set 2"),
          alpha_bootstrap = 1, text_to_show = NULL,
          cex_text = 5, col_text = "black", colour_bucket = FALSE,
          line_width = 0.5, col_JB_test = gray(.7), alpha_JB_test = .1)

moment_bucket_wrap(x, ..., weights = NULL, xvar = NULL, n_inter = 4,
          no_bootstrap = 99,
          col_bootstrap = hcl.colors(length.obj, palette = "Set 2"),
          alpha_bootstrap = 1, text_to_show = NULL,
          check_overlap_text = FALSE, cex_text = 5,
          col_text = "black", colour_bucket = FALSE,
          col_JB_test = gray(.7), alpha_JB_test = .1)
```

Arguments

`x`	x should be a continuous vector of a GAMLSS fitted model.
`...`	this for more that one continuous vectors or fitted models
`weights`	if priors weights are needed
`no_bootstrap`	the number of bootstrap samples for the cloud around the point of skewness and kurtosis.
`col_bootstrap`	The colour of the bootstrap samples
`alpha_bootstrap`	The transparency parameter of the bootstrap samples.
`text_to_show`	what text to show in the plots, default the names of vectors ot models
`cex_text`	the character size of the text
`col_text`	the colour of the text
`colour_bucket`	whether colour or gray lines in the bucket
`line_width`	the line width
`xvar`	the x term
`n_inter`	how many intervals needed
`check_overlap_text`	whether to check overlapping text
`col_JB_test`	the colour for the Jarque-Bera test
`alpha_JB_test`	the transparency constant for the Jarque-Bera test

Value

A plot displaying the transformed moment skewness and transformed moment kurtosis of the sample or residual of a model.

Note

The bucket plot provides an additional residual diagnostic tool that can be used for fitted model checking, alongside other diagnostic tools, for example worm plots, and Q (and Z) statistics.

Author(s)

Mikis Stasinopoulos, <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby <`r.rigby@londonmet.ac.uk`> and Fernanda De Bastiani

References

De Bastiani, F., Stasinopoulos, D. M., Rigby, R. A., Heller, G. Z., and Lucas A. (2022) Bucket Plot: A Visual Tool for Skewness and Kurtosis Comparisons. To be published.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/9780429298547 An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, doi:10.18637/jss.v023.i07.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M. D., Rigby, R. A., and De Bastiani F., (2018) GAMLSS: a distributional regression approach, *Statistical Modelling*, Vol. **18**, pp, 248-273, SAGE Publications Sage India: New Delhi, India.

(see also `https://www.gamlss.com/`).

See Also

`wp`, `Q.stats`

Examples

```
m1 <- gamlss(R~pb(Fl)+pb(A), data=rent, family=GA)
moment_bucket(m1)
moment_bucket_wrap(m1, xvar=rent$A)
```

`moment_gray_half` *Functions to create the background for the bucket plots*

Description

The functions plot the moment transformed skewness and moment transformed kurtosis of five important 4-parameter distributions in GAMLSS.

Usage

```
moment_gray_half(legend = FALSE)

moment_gray_both(line_width = 1)

moment_colour_half(legend = TRUE)

moment_colour_both(legend = TRUE, line_width = 1)

centile_colour_half(type = c("tail", "central"), legend = TRUE,
                   line_width = 1)

centile_colour_both(type = c("tail", "central"), legend = TRUE,
                   line_width = 1)

centile_gray_both(type = c("tail", "central"), legend = TRUE,
                  line_width = 0.5)
```

Arguments

`legend`	whether legend is required
`line_width`	line width
`type`	whether to plot 'tail' or 'central' skewness and kurtosis.

Details

The functions are described in Rigby *et al* (2019)

Value

A plot is created.

Note

The functions are use by the bucket plot function `model_mom_bucket()` to create the background of the bucket plots.

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda de Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/9780429298547 An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, doi:10.18637/jss.v023.i07.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

(see also `https://www.gamlss.com/`).

See Also

`momentSK`, `centileSK`

Examples

```
moment_gray_half()
moment_gray_both()
moment_colour_half()
moment_colour_both()

centile_colour_both()
centile_gray_both()
centile_colour_half()
```

`pcr_coef_path` *Plotting the fitted path of a PCR model.*

Description

This function is similar to the function `plot.PCR()` which is used to plot the path of a fitted principal componet regression model, fitted using the function `fitPCR()` of the package **gamlss.foreach**.

Usage

```
pcr_coef_path(x, legend=FALSE, plot=TRUE)

pcr_path(x, parameter = c("mu", "sigma", "nu", "tau"),
           legend = FALSE, plot = TRUE)
```

Arguments

`x`	a fitted PCR object (or a fitted GAMLSS object for function `pcr_path(`
`legend`	whether legent is needed
`plot`	whether to plot the path
`parameter`	which GAMLSS parameter, between "mu", "sigma", "nu", "tau"

Value

A gg-plot.

Author(s)

Mikis Stasinopoulos `<d.stasinopoulos@londonmet.ac.uk>`, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`plot.PCR`

Examples

```
library(gamlss.foreach)
library(glmnet)
data(QuickStartExample)
attach(QuickStartExample)
hist(y, main="(a)")
if (is.null(rownames(x)))
  colnames(x) <- paste0("X", seq(1:dim(x)[2]))
# fitting
MM<- fitPCR(x,y, k=log(100))
pp<-pcr_coef_path(MM)
pp+ geom_vline(xintercept = MM$pc, colour = "gray")
# using gamlss
m1 <- gamlss(y~pcr(x=x))
pcr_path(m1)
```

`pe_param` *Partial Effect of a term on the parameters and predictors*

Description

The function `pe_param()` is similar to the function `getPEF()` of the **gamlss** package. It plot the partial effect that a particular term has one of the parameters of the distribution or its predictor `eta`. The function `pe2_param()` is build for partial effects for from two terms and it is suitable to display first order interactions.

Usage

```
pe_param(obj = NULL, term = NULL, data = NULL, n.points = 100,
               parameter = c("mu", "sigma", "nu", "tau"),
               type = c("parameter", "eta"), scenario = list(),
               how = c("median", "last", "fixed"),
               col = "darkblue", size = 1.3, name.obj = NULL,
               rug.plot = TRUE, rug.col = "gray", rug.size = 0.5,
               data.plot = FALSE, data.col = "lightblue",
               data.size = 0.1, factor.size = 15,
               data.alpha = 0.9, bins = 30,
               filled = FALSE, ylim = NULL,
                    title)

pe_1_param(obj = NULL, term = NULL, data = NULL, n.points = 100,
                parameter = c("mu", "sigma", "nu", "tau"),
                type = c("parameter", "eta"),
                how = c("median", "last", "fixed"),
                scale.from = c("mean", "median", "none"),
                scenario = list(), col = "darkblue", size = 1.3,
                name.obj = NULL, data.plot = FALSE,
                data.col = "lightblue",data.size = 0.1,
                data.alpha = 0.9, rug.plot = TRUE, rug.col = "gray",
                rug.size = 0.5, factor.size = 15,
                ylim = NULL, title)

pe_2_param(obj = NULL, terms = NULL, data = NULL, n.points = 100,
                parameter = c("mu", "sigma", "nu", "tau"),
                type = c("parameter", "eta"),
                how = c("median", "last", "fixed"),
                scenario = list(), col = "darkblue",
                size = 1.3, data.plot = TRUE,
                data.col = "lightblue", data.size = 0.1,
                data.alpha = 0.9,bins = 30,
                filled = FALSE, name.obj = NULL, title)

pe_param_grid(model, terms, maxcol = 2, maxrow = 3, ylim=NULL, ...)
```

Arguments

`obj`	a GAMLSS fitted object
`model`	a GAMLSS fitted model

`term`	the model term we want to investigate can be one i.e. "Fl" or two c("Fl", "A")
`terms`	a list of model terms for example `list(c("Fl","A"), "H", "loc" ))`
`data`	the data frame used otherwise it takes it from the fitted model
`n.points`	the number of points for the evaluation of the term
`parameter`	the distribution parameter in which the term is fitted
`type`	here you specify or the distribution parameter i.e `"parameter"` or its prediction, `"eta"`
`how`	how to set all the other terms in the model
`scenario`	this can be a list of values for the rest of the terms in the model for the distribution parameter
`plot`	whether to plot the result
`col`	the colour of the partial effect of the term
`size`	the size of the line of partial effect of the term
`bins`	the number of binds for the contour plot
`filled`	whether to display the values in the contour
`title`	the title if different from the default
`name.obj`	this is a way to pass the name of the object
`maxcol`	the maximum columns in the grid plot
`maxrow`	the maximum rowss in the grid plot
`data.plot`	whether to plot the data
`rug.plot`	whether to print the rug bellow the figure
`rug.size`	the size of the rug
`rug.col`	the colour of the rug
`data.col`	the color of the data points
`data.size`	the size of the data points
`data.alpha`	the trnsparance constant of the data points
`factor.size`	the size of the symbol if a factor is plotted
`ylim`	if a common y limit is required
`scale.from`	whethet to substact from the mean the median or from zero
`...`	for passing argument from the function `pe_param_grid` to the function `pe_param`

Details

The functions `pe_param()` and `pe_param_grid()` can be used to help the use the interpretation of a GAMLSS model. The functions `pe_param()` provides the partial effect of one or two terms of a specified parameter of the distribution while the rest of the terms in the model are set on specific values or scenarios. The function `pe_param()` calls `pe_1param()` if the argument terms is one i.e. "Fl" or the function `pe_2param()` if the terms are two i.e. c("Fl"","A"). The `pe_param_grid()` plots multiple plots specified by the list used in the `term` argument.

Similar functions are `pe_quantile()` and `pe_moment()`.

Value

It is plotting the partial effect or is producing the resulting function

Author(s)

Mikis Stasinopulos, Rober Rigby and Fernanda de Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`getPEF`

Examples

```
m1 <- gamlss(R~pb(Fl)+pb(A)+loc+H, data=rent, gamily=GA)
pe_param(m1, "A")
# equivalent to
# pe_1_param(m1, "A")

pe_param(m1, c("Fl","A"), filled=TRUE)
#equivalent to
#pe_2_param(m1, c("Fl","A"), filled=TRUE)

pe_param_grid(m1, list(c("Fl","A"), c("H","loc")), filled=TRUE)
# the terms are additive no interaction
```

`resid_density` *Density of the residuals in a GAMLLSS model*

Description

The function `resid_density()` plots an histogram and a density estimator of the normalised quantile residuals from a fitted GAMLSS model. The function `model_density()` plots density estimators of the normalised quantile residuals from more than one fitted GAMLSS models.

Usage

```
resid_density(obj, resid, hist.col = "black", hist.fill = "white",
              dens.fill = "#FF6666", title)
model_density(obj, ..., title)
```

Arguments

`obj`	The function needs a GAMLSS fitted model or
`resid`	any standarised residual
`hist.col`	The colour of the border of the histogram
`hist.fill`	The colout of the hisogram
`dens.fill`	the colour of the desnsity
`title`	A title if needed
`...`	extra GAMLSS models

Details

This function `resid_density()` is a denity plot (similar to of the four plots produded when the plotting function `plot.gamlss()` is used within the **gamlss** package. I uses plotting function from the **ggplot2** package.

Value

A density plot of the residuals is produced.

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`plot.gamlss`

Examples

```
data(abdom)
a<-gamlss(y~pb(x),family=LO,data=abdom)
b<-gamlss(y~pb(x),family=NO,data=abdom)
resid_density(a)
model_density(a,b)
```

`resid_dtop` *Detrended Transformed Owen's Plot and ECDF for the residuals*

Description

The function `resid_dtop()` provides single detrended transformed Owen's plot, Owen (1995), for a GAMLSS fitted objects or any other residual vector (suitable standardised). This is a diagnostic tool for checking whether the normalised quantile residuals are coming from a normal distribution or not. This could be true if the horizontal line is within the confidence intervals.

The function `resid_ecdf()` provides the empirical cumulative distribution function of the residuals.

The function `y_ecdf()` provides the empirical cumulative distribution function of any numerical vectot `y`.

Usage

```
resid_dtop(obj, resid, type = c("Owen", "JW"), conf.level = c("95", "99"),
           value = 2, points.col = "steelblue4",
           check_overlap = TRUE,  title, ylim, ...)

resid_ecdf(obj, resid, type = c("Owen", "JW"), conf.level = c("95", "99"),
           value = 2, points.col = "steelblue4",
           check_overlap = TRUE,  show.outliers = TRUE, title, ...)

y_ecdf(y, type = c("Owen", "JW"), conf.level = c("95", "99"), value = 2,
           points.col = "steelblue4", check_overlap = TRUE,
           show.outliers = FALSE, title, ...)
```

Arguments

`obj`	A GAMLSS fitted model
`resid`	if the object is not specified the residual vector can be given here
`y`	a numeric vector
`type`	whether to use Owen (1995) or Jager and Wellner (2004) approximate formula
`conf.level`	95% (default) or 99% percent confidence interval for the plots
`value`	cut of point for large residuals
`points.col`	the colour of the points in the plot

`check_overlap`	to check for overlap when plotting the observation numbers
`title`	required title
`show.outliers`	whether to shoe the number of an outlier obsrvation
`ylim`	if the y limit should be different from the default max(y)+.1
`...`	further arguments

Value

A ggplot is generated

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda de Bastiani

References

Jager, L. and Wellner, J. A (2004) A new goodness of fit test: the reversed Berk-Jones statistic, University of Washington, Department of Statistics, Technical report 443. Owen A. B. (1995) Nonparametric Confidence Bands for a Distribution Function. Journal of the American Statistical Association Vol. 90, No 430, pp. 516-521.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), Appl. Statist., 54, part 3, 1-38.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`resid_wp`

Examples

```
data(abdom)
a<-gamlss(y~pb(x),sigma.fo=~pb(x,1),family=LO,data=abdom)
resid_dtop(a)
resid_ecdf(a)+ stat_function(fun = pNO, args=list(mu=0, sigma=1))
# create a gamma distributed random variable
y <- rGA(1000, mu=3, sigma=1)
gp<- y_ecdf(y)
gp + stat_function(fun = pGA, args=list(mu=3, sigma=1))
```

`resid_index` *A residual plots*

Description

The function `resid_index()` is plotting the residuals of a GAMLSS fitted model (or any other suitable standardised residual) against the observation number index.

The function `resid_mu()` plots the residuals against fitted values for `mu`.

The function `resid_median()` plots the residuals against fitted median values.

The function `resid_param()` plots the residuals against any of the GAMLSS fitted parameters, `mu`, `sigma`, `nu`, or `tau`.

The function `resid_quantile()` plots the residuals against any fitted quantile.

The function `resid_xvar()` plots the residuals against an explanatory term.

The function `resid_plots()` produces a plot similar to the one that the function `plot()` produce for a GAMLSS model in package **gamlss**. This is, four plots: a) `resid_index()`(b) `resid_mu()`, (c) `resid_density()` and (d) `resid_qqplot()`.

Residuals above (or below) certain specified value are identified.

Usage

```
resid_index(obj, resid, plot = TRUE, value = 2, title, annotate = TRUE,
           no.lines = FALSE)

resid_mu(obj, resid, plot = TRUE, value = 2, title, annotate = TRUE)

resid_median(obj, resid, plot = TRUE, value = 3, title,
            annotate = TRUE)

resid_param(obj, param = c("mu", "sigma", "nu", "tau"), title,
            line.col = "darkred", point.col = "steelblue4",
            point.shape = 20)

resid_quantile(obj, quantile = 0.5, title, newdata,
            line.col = "darkred", point.col = "steelblue4",
            point.shape = 20)

resid_plots(obj, theme = c("original", "ts", "new", "ecdf"), value = 3)

resid_xvar(obj, xvar, plot = TRUE, value = 2, title, annotate = TRUE)
```

Arguments

`obj`	a GAMLSS object
`resid`	or any other suitable standardised residual vector.

`xvar`	a continuous explanatory variable
`plot`	whether to plot the result
`param`	which GAMLSS parameter `mu`, `sigma`, `nu`, or `tau`
`value`	the cut off value for the identification of very large or very small residuals
`annotate`	whether the threshold annotation should appear or not
`line.col`	the colour of the line
`point.col`	the colour of the points
`point.shape`	the shape of the points
`title`	a title of the plot if needed
`theme`	what type of plots should `resid_plots()` used : "original" is like using `plot.gamlss()`, "ts" is like using `plot.gamlss(,ts="TRUE")` (not implemented yet), "new" it uses (a) `resid_index()`, (b) `resid_density()`, (c) `resid_wp()` and (d) `resid_dtop()`.
`no.lines`	this option allows to hide the horizontal lines so the resulting gg-plot can be used later with say `facet_wrap()` see example
`newdata`	whether the evaluation should be in newdata or the old data points
`quantile`	which quantile? default the median (0.50).

Value

A plot of the residuals is returned.

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC, doi:10.1201/9780429298547. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, doi:10.18637/jss.v023.i07.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M. D., Rigby, R. A., and De Bastiani F., (2018) GAMLSS: a distributional regression approach, *Statistical Modelling*, Vol. **18**, pp, 248-273, SAGE Publications Sage India: New Delhi, India.

(see also `https://www.gamlss.com/`).

See Also

`gamlss`, `plot.gamlss`

Examples

```
data(rent)
r1<-gamlss(R~pb(Fl)+pb(A)+H+loc,family=GA,data=rent)
resid_index(r1)
resid_mu(r1)
resid_median(r1)
resid_param(r1)
resid_quantile(r1)
resid_xvar(r1, A)
resid_plots(r1)
resid_index(r1, no.lines=TRUE)+facet_wrap(~ cut_number(rent$A, 6))
```

`resid_qqplot` *QQ-plot of the residuals of a GAMLSS model*

Description

The function `resid_qqplot()` produces a single QQ-plot of the residuals from a fitted GAMLSS model or any other model with suitable standardised residuals.

The function `add_resid_qqplot()` takes a QQ-plot created by `resid_qqplot()` and adds another QQ-plot from a different fitted model.

The function `add_resid_qqplots()` takes different fitted models and creates QQ-plots for all of them.

Usage

```
resid_qqplot(obj, resid, value = 3, points.col = "steelblue4",
              line.col = "darkgray", check_overlap = TRUE, title)

add_resid_qqplot(gg, obj, value = 3, points.col = "sienna",
             line.col = "darkgray", check_overlap = TRUE, title)

model_qqplot(obj, ..., line.col = "steelblue4", title)
```

Arguments

`obj`	A GAMLLS fitted model or
`resid`	any other residual suitable standardised.
`gg`	a ggplot
`value`	A cut off value to identify large or small residuals
`points.col`	the colout of the points in the plot

`line.col`	the colout of the line in the plot
`check_overlap`	if observations are identify this reduvce the cluterring
`title`	a title if needed it
`...`	extra GAMLSS models

Details

This is a stanard QQ-plot but with the advadance of able to identify large or samll residuals

Value

A QQ-plotbis created

Author(s)

Mikis Stasinopoulos `<d.stasinopoulos@londonmet.ac.uk>`, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`plot.gamlss`

Examples

```
data(abdom)
a<-gamlss(y~pb(x),family=LO,data=abdom)
b<-gamlss(y~pb(x),family=NO,data=abdom)
gg <- resid_qqplot(a)
add_resid_qqplot(gg, b)
model_qqplot(a,b)
```

`resid_symmetry` *Symmetry plots*

Description

The functions `resid_symmetry()` and `y_symmetry()` plot symmtry plots for residuals and single variable, respectively.

Usage

```
resid_symmetry(model, title)

y_symmetry(y, title)
```

Arguments

`model`	A model which allows the function `resid()`
`y`	a single variable
`title`	A title for the plot if needed

Details

The function orders the data (or residuals) and takes the median minus the lower half and plot it against the upper half minus the median.

Value

The symmetry plot is produced.

Author(s)

Mikis Stasinopoulos `<d.stasinopoulos@londonmet.ac.uk>`, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

See Also

`resid_index`

Examples

```
y <- rBCT(1000, mu=3, sigma=.1, nu=-1, tau=5)
y_hist(y)
gg <- y_symmetry(y)
```

`resid_wp` *Worm plot using ggplot2*

Description

The function produces worm plot of the residuals of a fitted model. A worm plot is a de-trended normal QQ-plot so departure from normality is highlighted.

The function `plot_wp()` it is similar to the **gamlss** package function `wp()` when the argument `xvar` is not used.

Usage

```
resid_wp(obj, resid, value = 3, points_col = "steelblue4",
         poly_col = "darkred",
         check_overlap = TRUE, title, ylim)

model_wp(obj, ..., title)

resid_wp_wrap(obj, resid, value = 3, xvar = NULL, n_inter = 4,
         points_col = "steelblue4", poly_col = "darkred",
         alpha_bound = 0.1, check_overlap = TRUE, title, ylim)

model_wp_wrap(obj, ..., xvar = NULL, value = 3, n_inter = 4,
         points_col = "steelblue4", alpha_bound = 0.1,
         check_overlap = TRUE, ylim, title)
```

Arguments

`obj`	a GAMLSS fitted object or any other fitted model where the `resid()` method works (preferably the residuals should be standardised or better normalised quantile residuals. Note for `model_wp` only gamlss object are accepted.)
`resid`	if object is missing this argument can be used to specify the residual vector (again it should a normalised quantile residual vector)
`value`	A cut off point to indicate large residuals, default is `value=3`
`xvar`	the x term for which the worm plot will be plotted against
`n_inter`	the number of intervals for continuous x-term

`points_col`	the color of the points in the plot
`poly_col`	the colour of the fitted polynomial in the plot
`check_overlap`	to check for overlap when plotting the observation numbers
`alpha_bound`	the transparency parameter for the coinfidence bound
`title`	required title
`ylim`	if the y limit should be different from the default max(y)+.1
`...`	extra GAMLSS models

Value

A worm plot is produced

Author(s)

Mikis Stasinopoulos <`d.stasinopoulos@londonmet.ac.uk`>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also `https://www.gamlss.com/`).

van Buuren and Fredriks M. (2001) Worm plot: simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, **20**, 1259–1277

See Also

`wp`

Examples

```
data(abdom)
# with data
a<-gamlss(y~pb(x),sigma.fo=~pb(x,1),family=LO,data=abdom)
resid_wp(a)
resid_wp(resid=resid(a))
```

`resp_mu` *Plotting the response against quantities of the fitted model*

Description

All plots are of the response variable against fitted values of interest.

The function `resp_mu()` is the ‘original’ one plotting the response against the parameter `mu`. The function reports the Pearson’s correlation coefficient and plot a lines (45% degrees) throught the graph.

The function `resp_param()` plots the response against any fitted parameter `mu`, `sigma`, `nu` or `tau`. The function also plots a smooth curve going throught the data and gives the Pearson’s correlation coefficient.

The function `resp_quantile()` plots the response against any fitted quantile, with default the median (0.50). The function also plots a smooth curve going throught the data and gives the Pearson’s correlation coefficient.

The function `quantile_gamlss()` is used by the function `resp_quantile()` to calculate the quantiles of the fitted distribution .

Usage

```
resp_mu(obj, title, line.col = "darkred",
           point.col = "steelblue4",
           point.shape = 20)

resp_param(obj, param = c("mu", "sigma", "nu", "tau"),
           title, line.col = "darkred", point.col = "steelblue4",
           point.shape = 20)

resp_quantile(obj, quantile = 0.5, title, newdata,
           line.col = "darkred", point.col = "steelblue4",
           point.shape = 20)

quantile_gamlss(obj, quantile = 0.5, newdata)
```

Arguments

`obj`	a GAMLSS fitted object
`param`	which parameters? `mu`, `sigma`, `nu` or `tau`, [only for `resp_param()`].
`quantile`	which quantile? default the median (0.50), [only for `resp_quantile()` and `quantile_gamlss()`].
`title`	a tittle if needed it, by default for the function `fitted_resp` it print the correlation coefficients between the two variable.
`line.col`	the colour of the line
`point.col`	the colour of the points

`point.shape`	the shape of the points
`newdata`	whether the evaluation should be in newdata or the old data points [only for functions `resp_quantile()` and `quantile_gamlss()`]

Details

This is standard plot in regression where the fitted values are plotted against the response. In GAMLSS model is done by plotting the response against the fitted values of the `mu` model which is most case is a location parameter.

Value

A plot is returned

Note

Do not use this plot if `mu` is not a location parameter.

Author(s)

Mikis Stasinopulos, Rober Rigby and Fernanda de Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also https://www.gamlss.com/).

See Also

`resid_plots`

Examples

```
m1 <- gamlss(R~pb(Fl)+pb(A)+H+loc, data=rent, family=GA )
resp_mu(m1)
resp_param(m1)
resp_quantile(m1)
```

---

y_hist *Histogram and density plot.*

Description

The function `y_hist()` creates a histogram and a density plot for a continuous variable.

The functions `y_acf()` and `y_pacf()` are autocorrolation and partial autocorrolation functions.

The function `y_dots()` is design for long tail right skewed variables. It is a plot emphasising the right tail of the distribution for such variables.

Usage

```
y_hist(y, data, with.density = TRUE, hist.col = "black",
         hist.fill = "white", dens.fill = "#FF6666",
         binwidth = (max(y)-min(y))/20,  from, to, title)

y_acf(x, data, title)

y_pacf(x, data, title)

y_dots(y, data, point.size = 2, point.col = "gray",
       line.col = c("red", "red"), line.type = c("solid", "dotted"),
       line.size = c(1, 1), x.axis.col = "black",
       x.axis.line.type = "solid", seed = 123, from, to, title)
```

Arguments

`y,x`	a continuous variable
`data`	where to find y
`with.density`	whether a density is required, default is `TRUE`
`hist.col`	the colour of lines in the histogram
`hist.fill`	the colour of the histigram
`dens.fill`	the color of the density plot
`binwidth`	for the histogram
`from`	where to start the histogram (you may have to change `binwidth`)
`to`	where to finish the histogram (you may have to change `binwidth`)
`point.size`	the size of the points in `y_dots`
`point.col`	the colour of the points in `y_dots`
`line.col`	the color of the verical lines indication the .50 and .90 quantiles in `y_dots`
`line.type`	the type of the verical lines indication the .50 and .90 quantiles in `y_dots`
`line.size`	the size of the verical lines indication the .50 and .90 quantiles in `y_dots`

`x.axis.col`	the colour of the x-axis
`x.axis.line.type`	the type of the x-axix
`seed`	the seed to jitter the y
`title`	use this for a different title

Value

A ggplot is returned

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

(see also https://www.gamlss.com/).

van Buuren and Fredriks M. (2001) Worm plot: simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, **20**, 1259–1277

See Also

`plot.ecdf`

Examples

```
y <- rBCT(1000, mu=3, sigma=.1, nu=-1, tau=5)
y_hist(y)
gg <- y_hist(y, with.dens=FALSE)
gg+stat_function(fun = dBCT, args=list(mu=3, sigma=.1,  nu=-1, tau=5),
                 colour = "black")
gg+ stat_function(fun = dBCT, args=list(mu=3, sigma=.1,  nu=-1, tau=5),
                  geom = "area", alpha=0.5, fill="pink", color="black", n=301)

y_acf(diff(EuStockMarkets[,1]))

y_dots(rent$R)
```

Index